Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 374 (2007) 82–90
Theoretical Computer Science
www.elsevier.com/locate/tcs

On the (n, t)-antipodal Gray codes

Gerard J. Chang[a,b,c], Sen-Peng Eu[d,*], Chung-Heng Yeh[e]

[a] *Department of Mathematics, National Taiwan University, Taipei 10617, Taiwan*
[b] *Taida Institute for Mathematical Sciences, National Taiwan University, Taipei 10617, Taiwan*
[c] *National Centre for Theoretical Sciences, Taipei Office, Taiwan*
[d] *Department of Applied Mathematics, National University of Kaohsiung, Kaohsiung 81148, Taiwan*
[e] *Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan*

Received 30 January 2006; received in revised form 21 November 2006; accepted 7 December 2006

Communicated by D.-Z. Du

Abstract

An n-bit Gray code is a circular listing of all 2^n n-bit binary strings in which consecutive strings differ at exactly one bit. For $n \le t \le 2^{n-1}$, an (n, t)-antipodal Gray code is a Gray code in which the complement of any string appears t steps away from the string, clockwise or counterclockwise. Killian and Savage proved that an (n, n)-antipodal Gray code exists when n is a power of 2 or $n = 3$, and does not exist for $n = 6$ or odd $n > 3$. Motivated by these results, we prove that for odd $n \ge 3$, an (n, t)-antipodal Gray code exists if and only if $t = 2^{n-1} - 1$. For even n, we establish two recursive constructions for (n, t) codes from smaller (n', t'). Consequently, various (n, t)-antipodal Gray codes are found for even n's. Examples are for $t = 2^{n-1} - 2^k$ with k odd and $1 \le k \le n - 3$ when $n \ge 4$, for $t = 2^{n-k}$ when $n \ge 2k$ with $1 \le k \le 3$, for $t = n$ when $n = 2^k \ge 2$ (an alternative proof for Killian and Savage's result) ... etc.

Keywords: Antipodal; Gray code; Hamiltonian cycle; n-cube; Complement; Period

1. Introduction

Given a positive integer n, an *n-bit Gray code* (*Gray code* for short) is a circular listing of all 2^n n-bit binary strings, in which two consecutive strings differ at exactly one bit [10].

Gray codes have wide applications such as graphics and image processing [1], information storage and retrieval [3], processor allocation in the hypercube [4], statistics [5], hashing [6], puzzle [7], ordering of documents on shelves [13], signal encoding [14], computing the permanent [15], data compression [17], and circuit testing [18]. An extensive bibliography lists hundreds of references in the survey paper by Savage [19].

There are many ways to generate a Gray code for any given n. A classic recursive approach constructs an n-bit Gray code by appending 0 to each string of the $(n - 1)$-bit Gray code, then list the $(n - 1)$ Gray code in reverse, appending 1 to each string. A Gray code generated in this way is called a *reflected Gray code*. However, Gray codes

* Corresponding author. Tel.: +886 75919526.
E-mail addresses: gjchang@math.ntu.edu.tw (G.J. Chang), speu@nuk.edu.tw (S.-P. Eu), chungheng.yeh@gmail.com (C.-H. Yeh).

0304-3975/$ - see front matter
doi:10.1016/j.tcs.2006.12.005

A (4,4)−antipodal Gray code	0000	0001	0011	0111	1111	1110	1100	1000
	1010	1011	1001	1101	0101	0100	0110	0010
A (4,6)−antipodal Gray code	0000	0001	0011	0010	0110	0111	1111	1110
	1010	1011	1001	1000	1100	1101	0101	0100
A (4,8)−antipodal Gray code	0000	0001	0011	0010	0110	0100	0101	0111
	1111	1110	1100	1101	1001	1011	1010	1000

Fig. 1. Examples of $(4, t)$-antipodal Gray codes for $t = 4, 6, 8$.

with certain additional properties or restrictions would be desirable. In fact, there are many variations other than the reflected Gray codes. The *balanced* Gray codes, *monotone* Gray codes, *non-composite* Gray codes, *long-run* Gray codes are among some of these [2,9,20,21]. The reader may refer to [12] and [19] for more information.

In the language of graph theory, a Gray code is equivalent to a Hamiltonian cycle of an *n-cube*, the graph whose vertices are the n-bit binary strings with two vertices being adjacent if they differ at exactly one bit [8,16].

Recently Killian and Savage [11] investigated the *antipodal Gray code*, which is a Gray code with the restriction that the complement of any string appears at exactly n strings away, either counterclockwise or clockwise. They proved that the antipodal Gray code does not exist if $n > 3$ is odd or $n = 6$. They also gave a clever construction for the existence of an antipodal Gray code for n being a power of 2. Motivated by their results, we study antipodal Gray codes in a more general setting as follows.

For positive integers $n \leq t \leq 2^{n-1}$, an (n, t)*-antipodal Gray code* is a Gray code in which the complement of any string appears t steps away from the string, either clockwise or counterclockwise. An antipodal Gray code is then an (n, n)-antipodal Gray code. See Fig. 1 for examples of $(4, t)$-antipodal Gray codes for $t = 4, 6, 8$. Notice that for the existence of an (n, t)-antipodal Gray code it is necessary that n and t have the same parity.

This paper is organized as follows. Notation and preliminary results are given in Section 2. In Section 3, we characterize t for which an (n, t)-antipodal Gray code exists when $n \geq 3$ is odd. More precisely, it is proved that for odd $n \geq 3$ there is an (n, t)-antipodal Gray code if and only if $t = 2^{n-1} - 1$. In Section 4 we consider the case when n is even. In particular, we establish two recursive constructions for (n, t) codes from smaller (n', t'). Consequently, various (n, t)-antipodal Gray codes are found for even n's. Examples are for $t = 2^{n-1} - 2^k$ with k odd and $1 \leq k \leq n - 3$ when $n \geq 4$, for $t = 2^{n-k}$ when $n \geq 2k$ with $1 \leq k \leq 3$, for $t = n$ when $n = 2^k \geq 2$ (an alternative proof for Killian and Savage's result [11]) . . . etc. The final section discusses some unsolved problems.

2. Notation and preliminary results

In this paper, we employ terminology and notation used in [11] by modifying (n, n) to (n, t). Results in Lemmas 1–8 are parallel to those in [11]. However, for clearness and for readers' convenience we include the proofs in this paper. Also, some of our constructions share the similar spirit as those in [11].

An n-bit *Gray code* G is a circular listing of all 2^n n-bit binary strings, in which consecutive strings differ at exactly one bit. We denote G by $G := (G(0), G(1), \ldots, G(2^n - 1))$, where $G(i)$ and $G(i + 1)$, with indices module 2^n, are consecutive. The *complement* of a bit is defined by $\overline{0} = 1$ and $\overline{1} = 0$. The *complement* of a binary string $x_1x_2 \ldots x_n$ is defined by $\overline{x} = \overline{x_1}\,\overline{x_2} \ldots \overline{x_n}$.

An (n, t)*-antipodal Gray code* is an n-bit Gray code with the additional property that the complement of each string is t steps away, either counterclockwise or clockwise. More precisely, for $n \leq t \leq 2^{n-1}$, G is an (n, t)*-antipodal Gray code* if for $0 \leq i < 2^n$, we have:

$$\text{(i) } \overline{G(i)} = G(i \oplus t) \quad \text{or} \quad \text{(ii) } \overline{G(i)} = G(i \oplus -t),$$

where $i \oplus j := (i + j) \bmod 2^n$ for integers i and j. It is clear that $\oplus$ is commutative and associative. We pair up each string with its complement: the index i is called a *pre-element* in case (i), and is called a *post-element* in case (ii). Note that an element is both a pre-element and a post-element if and only if $t = 2^{n-1}$. For example, for the (4, 6)-antipodal Gray code in Fig. 1, indices 0, 1, 4, 5, 8, 9, 12, 13 are the pre-elements while 2, 3, 6, 7, 10, 11, 14, 15 the post-elements.

As the cases when $n = 1, 2$ are easy, from now on, we may consider (n, t)-antipodal Gray codes only for $3 \le n \le t \le 2^{n-1}$ and $n \equiv t \pmod 2$. First, we present some basic lemmas.

Lemma 1. *For $0 \le i < 2^n$, index i is a pre-element if and only if $i \oplus t$ is a post-element.*

Proof. If i is a pre-element, then $\overline{G(i)} = G(i \oplus t)$. This gives $\overline{G(i \oplus t)} = G(i) = G((i \oplus t) \oplus -t)$ and so $i \oplus t$ is a post-element. Conversely, If $i \oplus t$ is a post-element, then $\overline{G(i \oplus t)} = G((i \oplus t) \oplus -t) = G(i)$, which gives $\overline{G(i)} = G(i \oplus t)$ and so i is a pre-element. □

When $n \le t < 2^{n-1}$, the *sign sequence* of G is defined as $\sigma_G := (\sigma_G(0), \sigma_G(1), \ldots, \sigma_G(2^n - 1))$, where $\sigma_G(i) =$ '+' if i is a pre-element and $\sigma_G(i) =$ '−' if i is a post-element. For the case of $t = 2^{n-1}$, we let $\sigma_G(i) =$ '+' if $0 \le i < 2^{n-1}$ and $\sigma_G(i) =$ '−' otherwise. For example, the $(4, 6)$ code in Fig. 1 has $\sigma_G(i) = (+, +, -, -, +, +, -, -, +, +, -, -, +, +, -, -)$ as the sign sequence.

Lemma 2. *For $0 \le i < 2^n$, we have $\sigma_G(i) = \sigma_G(i \oplus 2t)$.*

Proof. According to Lemma 1, i is a pre-element if and only if $i \oplus t$ is a post-element, which happens if and only if $i \oplus 2t$ is a pre-element, hence $\sigma_G(i) = \sigma_G(i \oplus 2t)$. □

An integer p is a *period* for σ_G if $\sigma(i) = \sigma_G(i \oplus p)$ for $0 \le i < 2^n$. It is clear that 2^n is a period, $2t$ is a period by Lemma 2, and t is not a period by Lemma 1. We compute the minimum positive period of a sign sequence in next lemma.

Lemma 3. *The minimum positive period of σ_G is* $\gcd(2^n, 2t)$.

Proof. Let $p = \gcd(2^n, 2t)$. There exist integers a and b such that $p = a \cdot 2^n + b \cdot 2t$, which implies that p is also a period since 2^n and $2t$ are periods. Suppose p^* is the minimum positive period. By the same argument, $\gcd(p^*, p)$ is also a period. As $\gcd(p^*, p) \le p^*$, we have that $p^* = \gcd(p^*, p)$ which is a divisor of p and so is of the form 2^c for some positive integer c. If $p^* < p \le 2t$, then p^* is a divisor of t and so t is a period, which contradicts Lemma 1. Thus, $p^* = p$. □

We may record the position of bit change by defining the *flip sequence* f_G of G by letting $f_G(i)$ be the bit position at which $G(i)$ and $G(i \oplus -1)$ differ for $0 \le i < 2^n$. For example, the flip sequence of the $(4, 6)$-antipodal Gray code in Fig. 1 is $f_G = (3, 4, 3, 4, 2, 4, 1, 4, 2, 4, 3, 4, 2, 4, 1, 4)$. With this we can characterize the sign sequence:

Lemma 4. *$\sigma_G(i) =$ '+' if and only if each element in $\{1, 2, \ldots, n\}$ appears an odd number of times in the multiset $F_i := \{f_G(i \oplus k) : 1 \le k \le t\}$.*

Proof. The lemma follows from the fact that $\sigma_G(i) =$ '+' if and only if $G(i)$ and $G(i \oplus t)$ differ at all n positions i, which occurs if any only if G flips at each position j an odd number of times. □

3. (n, t)-antipodal Gray code for odd n

In this section we consider the case when n is odd. Our main result is to prove that for any odd integer $n \ge 3$ there is an (n, t)-antipodal Gray code if and only if $t = 2^{n-1} - 1$. Meanwhile we also construct an $(n, 2^{n-1})$-antipodal Gray code for any positive even integer n.

Lemma 5. *If G is an (n, t)-antipodal Gray code with odd $n \ge 3$, then the minimum positive period for σ_G is 2. That is, $\sigma_G = (+, -, +, -, \ldots, +, -)$.*

Proof. Since $n \equiv t \pmod 2$ and n is odd, t is odd. By Lemma 3, the minimum positive period is $\gcd(2^n, 2t) = 2$. □

In the next few lemmas we establish properties for f_G.

Lemma 6. *If G is an (n, t)-antipodal Gray code with odd $n \ge 3$ and $\sigma_G(i) =$ '+', then $f_G(i \oplus 1 \oplus t) \ne f_G(i \oplus 1)$.*

Proof. Suppose to the contrary that $f_G(i \oplus 1 \oplus t) = f_G(i \oplus 1)$. Then, $F_{i \oplus 1} = (F_i \setminus \{f_G(i \oplus 1)\}) \cup \{f_G(i \oplus 1 \oplus t)\} = F_i$. Since $\sigma_G(i) =$ '+', by Lemma 4, each $j \in \{1, 2, \ldots, n\}$ appears an odd number of times in $F_i = F_{i \oplus 1}$, hence again by Lemma 4 $\sigma_G(i \oplus 1) =$ '+', which contradicts Lemma 5. □

Lemma 7. *If G is an (n, t)-antipodal Gray code with odd $n \geq 3$ and $\sigma_G(i) =$ '+', then $f_G(i \oplus 1) = f_G((i \oplus 1) \oplus (t + 1))$ and $f_G(i \oplus 2) = f_G((i \oplus 2) \oplus (t - 1))$.*

Proof. Let $f_G(i \oplus 1) = x$ and $f_G(i \oplus 2) = y$. Clearly $x \neq y$. According to Lemma 5, $\sigma_G(i \oplus 2) = \sigma_G(i) =$ '+'. By Lemma 4, both x and y appear an odd number of times in F_i and $F_{i\oplus 2}$. So we must have that $\{f_G(i \oplus (t + 1)), f_G(i \oplus (t + 2))\} = \{x, y\}$. On the other hand, from Lemma 6 we know $x = f_G(i \oplus 1) \neq f_G(i \oplus (t + 1))$. Hence, $x = f_G(i \oplus 1) = f_G(i \oplus (t+2)) = f_G((i \oplus 1) \oplus (t+1))$ and $y = f_G(i \oplus 2) = f_G(i \oplus (t+1)) = f_G((i \oplus 2) \oplus (t-1))$, as desired. □

In fact, we can fully predict the position of the bit change as follows.

Lemma 8. *If G is an (n, t)-antipodal Gray code with odd $n \geq 3$ and $L = \mathrm{lcm}(t + 1, t - 1)$, then $f_G(i) = f_G(i \oplus L)$ for $0 \leq i < 2^n$.*

Proof. Since both $t+1$ and $t-1$ are even, for $\sigma_G(i) =$ '+', by Lemma 5 we have $\sigma_G(i \oplus r(t+1)) = \sigma_G(i \oplus r(t-1)) =$ '+' for all integers $r \geq 0$. Repeatedly applying Lemma 7 gives $f_G(i \oplus 1) = f_G((i + 1) \oplus r(t + 1))$ and $f_G(i \oplus 2) = f_G((i + 2) \oplus r(t - 1))$ for all r. In particular, if $\sigma_G(i) =$'+', then $f_G(i \oplus 1) = f_G((i \oplus 1) \oplus L)$ and $f_G(i \oplus 2) = f_G((i \oplus 2) \oplus L)$. Hence $f_G(i) = f_G(i \oplus L)$ for any i. □

We need two more lemmas for constructing (n, t)-antipodal Gray codes. We call a listing of 2^n binary strings an *n-Gray path* if any two consecutive strings differ at exactly one position, with the extra condition that the first string is 0^n and the last 1^n.

Lemma 9. *For any positive odd integer n, there is an n-Gray path.*

Proof. We shall prove the lemma by induction on n. The case of $n = 1$ is trivial. Suppose the lemma holds for $n = k$, where k is odd, say $(a_0, a_1, \ldots, a_{2^k-1})$ is a Gray path. Now consider the following sequence of 2^{k+2} distinct $(k + 2)$-bit binary strings, reading from left to right and then from top to bottom:

$$\begin{array}{cccccccc}
00a_0, & 01a_0, & 11a_0, & 10a_0, & 10a_1, & 11a_1, & 01a_1, & 00a_1, \\
00a_2, & 01a_2, & 11a_2, & 10a_2, & 10a_3, & 11a_3, & 01a_3, & 00a_3, \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
00a_{2^k-4}, & 01a_{2^k-4}, & 11a_{2^k-4}, & 10a_{2^k-4}, & 10a_{2^k-3}, & 11a_{2^k-3}, & 01a_{2^k-3}, & 00a_{2^k-3}, \\
00a_{2^k-2}, & 01a_{2^k-2}, & 11a_{2^k-2}, & 10a_{2^k-2}, & 10a_{2^k-1}, & 00a_{2^k-1}, & 01a_{2^k-1}, & 11a_{2^k-1}.
\end{array}$$

A straightforward check shows that this is a Gray path. □

Theorem 10. *For any positive even integer n, there is an $(n, 2^{n-1})$-antipodal Gray code.*

Proof. By Lemma 9, there is an $(n - 1)$-Gray path $(a_0, a_1, \ldots, a_{2^{n-1}-1})$. Consider the following sequence of 2^n distinct n-bit binary strings:

$$\begin{array}{cccc}
0a_0, & 0a_1, & \ldots, & 0a_{2^{n-1}-1}, \\
1\overline{a_0}, & 1\overline{a_1}, & \ldots, & 1\overline{a_{2^{n-1}-1}}.
\end{array}$$

It is easy to see that this is an $(n, 2^{n-1})$-antipodal Gray code. □

We are now ready to prove the main result of this section.

Theorem 11. *For any odd integer $n \geq 3$, there is an (n, t)-antipodal Gray code G if and only if $t = 2^{n-1} - 1$.*

Proof. Suppose there is an (n, t)-antipodal Gray code G. According to Lemma 8, $f_G(i \oplus k) = f_G(i \oplus k \oplus L)$ for $1 \leq k \leq L$, where $L = \mathrm{lcm}(t + 1, t - 1)$. Thus, between $G(i)$ and $G(i \oplus 2L)$, each position flips an even number of times, which gives that $G(i) = G(i \oplus 2L)$ for $0 \leq i < 2^n$. Since all elements of G are distinct, $2L$ is a multiple of 2^n. Now, $2L = 2 \cdot \mathrm{lcm}(t + 1, t - 1) = \gcd(t + 1, t - 1) \cdot \mathrm{lcm}(t + 1, t - 1) = (t + 1)(t - 1)$. Hence, either $t + 1$ or $t - 1$ is a multiple of 2^{n-1}, which is only possible when $t = 2^{n-1} - 1$.

Next, we construct an $(n, 2^{n-1} - 1)$-antipodal Gray code for any odd integer $n \geq 3$. By Theorem 10, there is an $(n-1, 2^{n-2})$-antipodal Gray code $(G(0), G(1), \ldots, G(2^{n-1}-1))$. Consider the following sequence of 2^n distinct n-bit binary strings:

$$\begin{array}{llll}
0G(0), & 1G(0), & 1G(1), & 0G(1), \\
\vdots & \vdots & \vdots & \vdots \\
0G(2^{n-2}-2), & 1G(2^{n-2}-2), & 1G(2^{n-2}-1), & 0G(2^{n-2}-1), \\
0G(2^{n-2}), & 1G(2^{n-2}), & 1G(2^{n-2}+1), & 0G(2^{n-2}+1), \\
\vdots & \vdots & \vdots & \vdots \\
0G(2^{n-1}-2), & 1G(2^{n-1}-2), & 1G(2^{n-1}-1), & 0G(2^{n-1}-1).
\end{array}$$

Notice that we can attach the prefix pattern 0, 1, 1, 0 properly as above since 2^{n-2} is even. As $xG(i)$ is the complement of $\overline{x}G(2^{n-2}+i)$ for $0 \leq i < 2^{n-2}$, a straightforward check shows that this is an $(n, 2^{n-1}-1)$-antipodal Gray code. □

4. (n, t)-antipodal Gray code for even n

This section is devoted to constructing (n, t)-antipodal Gray codes for positive even integers n. Two theorems for constructing (n, t)-antipodal Gray codes from (n', t')-antipodal Gray codes with $n' < n$ and $t' < t$ are established. Consequently, various (n, t)-antipodal Gray codes are found for even n's. Examples are for $t = 2^{n-1} - 2^k$ with k odd and $1 \leq k \leq n-3$ when $n \geq 4$, for $t = 2^{n-k}$ when $n \geq 2k$ with $1 \leq k \leq 3$, for $t = n$ when $n = 2^k \geq 2$ (an alternative proof for Killian and Savage's result [11]) . . . etc.

Theorem 12. *For any even integer $n \geq 4$ and any odd integer k with $1 \leq k \leq n-3$, there is an $(n, 2^{n-1} - 2^k)$-antipodal Gray code.*

Proof. Let $h = n - k \geq 3$, which is an odd integer. By Theorem 11, there is an $(h, 2^{h-1} - 1)$-antipodal Gray code $(G(0), G(1), \ldots, G(2^h - 1))$. Also by Lemma 9, there is a k-Gray path $(a_0, a_1, \ldots, a_{2^k-1})$. We now construct the following sequence of $2^k 2^h = 2^n$ distinct n-bit binary strings:

$$\begin{array}{llll}
a_0G(0), & a_1G(0), & \ldots, & a_{2^k-1}G(0), \\
\overline{a_0}G(1), & \overline{a_1}G(1), & \ldots, & \overline{a_{2^k-1}}G(1), \\
a_0G(2), & a_1G(2), & \ldots, & a_{2^k-1}G(2), \\
\overline{a_0}G(3), & \overline{a_1}G(3), & \ldots, & \overline{a_{2^k-1}}G(3), \\
\vdots & \vdots & \vdots & \vdots \\
a_0G(2^{h-1}-2), & a_1G(2^{h-1}-2), & \ldots, & a_{2^k-1}G(2^{h-1}-2), \\
\overline{a_0}G(2^{h-1}-1), & \overline{a_1}G(2^{h-1}-1), & \ldots, & \overline{a_{2^k-1}}G(2^{h-1}-1), \\
a_0G(2^{h-1}), & a_1G(2^{h-1}), & \ldots, & a_{2^k-1}G(2^{h-1}), \\
\overline{a_0}G(2^{h-1}+1), & \overline{a_1}G(2^{h-1}+1), & \ldots, & \overline{a_{2^k-1}}G(2^{h-1}+1), \\
\vdots & \vdots & \vdots & \vdots \\
a_0G(2^h-2), & a_1G(2^h-2), & \ldots, & a_{2^k-1}G(2^h-2), \\
\overline{a_0}G(2^h-1), & \overline{a_1}G(2^h-1), & \ldots, & \overline{a_{2^k-1}}G(2^h-1).
\end{array}$$

It is straightforward to check that this is an $(n, 2^{n-1} - 2^k)$-antipodal Gray code. □

In the following we shall construct (n, t)-antipodal Gray codes from (n', t')-antipodal Gray codes with $n' < n$ and $t' < t$. An (n, t)-antipodal Gray code G is called *simple* if $2t$ is a divisor of 2^n and $\sigma_G(i) =$'+' for $0 \leq (i \bmod 2t) < t$. In this case we also have $\sigma_G(i) =$ '−' for $t \leq (i \bmod 2t) < 2t$. Hence the sign sequence of a simple code consists of repeating a run of t '+'s by a run of t '−'s. Notice that we have $\overline{G(2tj + k)} = G(2tj + k + t)$ for $0 \leq j < 2^n/2t$ and $0 \leq k < t$. For an example, see the (4, 4)-antipodal Gray code in Fig. 1, which is simple with the sign sequence $(+, +, +, +, -, -, -, -, +, +, +, +, -, -, -, -)$.

Now, the first construction.

Theorem 13. *If n is a positive even integer and there is a simple (n, t)-antipodal Gray code, then there is a simple $(n + 2, 4t)$-antipodal Gray code.*

Proof. Suppose $(G(0), G(1), \ldots, G(2^n - 1))$ is a simple (n, t)-antipodal Gray code. For $0 \le j < 2^n/2t$, let X_j be the sequence of $4t$ distinct $(n + 2)$-bit binary strings:

$$\begin{array}{llll}
00G(2tj+0), & 01G(2tj+0), & 11G(2tj+0), & 10G(2tj+0),\\
10G(2tj+1), & 00G(2tj+1), & 01G(2tj+1), & 11G(2tj+1),\\
11G(2tj+2), & 01G(2tj+2), & 00G(2tj+2), & 10G(2tj+2),\\
10G(2tj+3), & 00G(2tj+3), & 01G(2tj+3), & 11G(2tj+3),\\
\vdots & & & \vdots\\
11G(2tj+t-2), & 01G(2tj+t-2), & 00G(2tj+t-2), & 10G(2tj+t-2),\\
10G(2tj+t-1), & 00G(2tj+t-1), & 01G(2tj+t-1), & 11G(2tj+t-1).
\end{array}$$

Also, let $\overline{X_j}$ be the sequence obtained from X_j by replacing each item by its complement. Then, it can be check that $(X_0, \overline{X_0}, X_1, \overline{X_1}, \ldots, X_{2^n/2t-1}, \overline{X_{2^n/2t-1}})$ is a simple $(n + 2, 4t)$-antipodal Gray code. □

Consequently, we have the following result.

Theorem 14. *Suppose n is a positive even integer and $1 \le k \le 3$. If $n \ge 2k$, then there is a simple $(n, 2^{n-k})$-antipodal Gray code.*

Proof. The theorem follows from Theorem 13, the simple (2, 2)-antipodal Gray code (00, 01, 11, 10), the simple (4, 4)-antipodal Gray codes in Fig. 1, and the following simple (6, 8)-antipodal Gray code generated by computer.

000000, 100000, 110000, 010000, 011000, 111000, 111100, 111110,
111111, 011111, 001111, 101111, 100111, 000111, 000011, 000001,
100001, 110001, 010001, 011001, 111001, 111101, 011101, 011100,
011110, 001110, 101110, 100110, 000110, 000010, 100010, 100011,
101011, 001011, 011011, 111011, 110011, 010011, 010111, 010101,
010100, 110100, 100100, 000100, 001100, 101100, 101000, 101010,
001010, 011010, 111010, 110010, 010010, 010110, 110110, 110111,
110101, 100101, 000101, 001101, 101101, 101001, 001001, 001000. □

As we will see, the simple codes will be the key for constructing longer codes. We need another notion to decompose a simple Gray code. An (n, t)-*segment*, denoted $(P_0, P_1, \ldots, P_{2t-1})$, is a listing of $2t$ consecutive strings of an (n, t)-antipodal Gray code, satisfying $P_i = \overline{P_{i+t}}$ for $0 \le i < t$. For example, in the (6, 8)-antipodal Gray code above there are four (6, 8)-segments, each consisting of 16 strings, namely those in lines 1 and 2, those in lines 3 and 4, those in lines 5 and 6, and those in lines 7 and 8. A simple (n, t)-antipodal Gray Code G is then the concatenation of $k = 2^n/2t$ segments. We denote this fact by $G = (N_1, N_2, \ldots, N_k)$, where each N_i, $1 \le i \le k$, is an (n, t)-segment, and call it the *segment decomposition* of G.

Lemma 15. *If $(P_0, P_1, \ldots, P_{2t-1})$ is an (n, t)-segment, then P_0 and P_{2t-1} differ at exactly one bit.*

Proof. Since P_{t-1} and P_t differ at exactly one bit, and $P_{2t-1} = \overline{P_{t-1}}$, we have that P_{2t-1} and P_t differ at $n - 1$ bits. We also know that P_0 and P_t differ at every bit, hence P_0 and P_{2t-1} differ at exactly one bit. □

From this lemma it can be easily deduced that the *reverse*, reading backward from the last string, or the *rotation*, reading from any string with the indices module $2t$, of an $(n.t)$-segment is also an (n, t)-segment.

Lemma 16. *We may construct $2t$ different $(n + m, 2t)$-segments from an (n, t)-segment and an (m, t)-segment.*

Proof. Suppose $(P_0, P_1, \ldots, P_{2t-1})$ is an (m, t)-segment and $(Q_0, Q_1, \ldots, Q_{2t-1})$ is an (n, t)-segment. Consider the following sequence of $4t$ distinct $(n + m)$-bit binary strings:

$$\begin{array}{llllllll}
P_0Q_0, & P_0Q_1, & P_1Q_1, & \ldots, & P_{t-2}Q_{t-2}, & P_{t-2}Q_{t-1}, & P_{t-1}Q_{t-1}, & P_{t-1}Q_t,\\
P_tQ_t, & P_tQ_{t+1}, & P_{t+1}Q_{t+1}, & \ldots, & P_{2t-2}Q_{2t-2}, & P_{2t-2}Q_{2t-1}, & P_{2t-1}Q_{2t-1}, & P_{2t-1}Q_0.
\end{array}$$

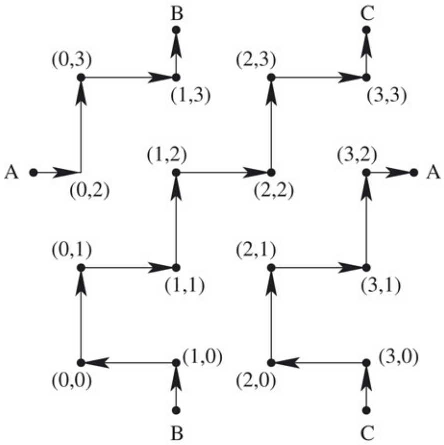

Fig. 2. Traverse strings of a block of the Cartesian product of two (2, 2)-segments.

It is easy to check that this is an $(n+m, 2t)$-segment. By rotating the sequence $(P_0, P_1, \ldots, P_{2t-1})$, we can construct $2t$ different $(n+m, 2t)$-segments. $\square$

We now come up to the main result of this section.

Theorem 17. *For any positive even integers m and n with t a divisor of both 2^m and 2^n, if there are a simple (m,t)-antipodal Gray code G_1 and a simple (n,t)-antipodal Gray code G_2, then there is a simple $(m+n, 2t)$-antipodal Gray code.*

Proof. The proof strategy is to use segment decomposition of $G_1 = (M_1, M_2, \ldots, M_p)$ and $G_2 = (N_1, N_2, \ldots, N_q)$, where $p = 2^m/2t - 1$ and $q = 2^n/2t - 1$. From these decompositions, we consider the Cartesian product $G_1 \times G_2$, which has *pq blocks*. By a block we mean the set $M_i \times N_j$, $1 \le i \le p$, $1 \le j \le q$. Each block consists of $4t^2$ strings of $(m+n)$ bits, with each string the first m bits coming from a string in M_i and the last n bits coming from a string in N_j. Our desired $(m+n)$-antipodal Gray code is obtained by traversing all blocks, together with the requirement that strings in the same block should be traversed before passing into the next block. This requirement is ensured below.

Take a block, which is a Cartesian product of an (m,t)-segment and an (n,t)-segment. We design a method to traverse all $4t^2$ strings of this block as follows. We select t disjoint $(n+m, 2t)$-segments constructed in Lemma 16. Let $i \uplus j := (i+j) \bmod 2t$ for integers i, j. For all $0 \le k < t$, let X_k be the following segment of $4t$ distinct $(n+m)$-bit binary strings:

$$\begin{array}{llllllll} P_{0\uplus -2k}Q_0, & P_{0\uplus -2k}Q_1, & P_{1\uplus -2k}Q_1, & \ldots, & P_{t-2\uplus -2k}Q_{t-1}, & P_{t-1\uplus -2k}Q_{t-1}, & P_{t-1\uplus -2k}Q_t, \\ P_{t\uplus -2k}Q_t, & P_{t\uplus -2k}Q_{t+1}, & P_{t+1\uplus -2k}Q_{t+1}, & \ldots, & P_{2t-2\uplus -2k}Q_{2t-1}, & P_{2t-1\uplus -2k}Q_{2t-1}, & P_{2t-1\uplus -2k}Q_0. \end{array}$$

It is straightforward to check that the last string of X_k and the first string of X_{i+k} differ at exactly one bit. In addition, the first string of X_0 and the last string of X_{t-1} also differ at exactly one bit. Thus, $(X_0, X_1, \ldots, X_{t-1})$ is a listing of $4t^2$ $(m+n)$-bit binary strings in which consecutive items differ at exactly one bit.

It is easier to grasp the idea if we regard the listing as a special directed Hamiltonian cycle on a torus. An illustrated example is in Fig. 2, where we traverse all strings of a block resulting from Cartesien product of two (2, 2)-segments. In this example (i, j), $0 \le i, j \le 3$, stands for the $(2+2)$-bit string with the first two bit from M_i and the last two bits from N_j, and (i, j) is adjacent to $(i, j \pm 1)$ and $(i \pm 1, j)$ with coordinates module $2t$. The listing (X_0, X_1) is a directed Hamiltonian cycle of this graph.

In this way we can traverse every string in a block. Moreover, there are many possibilities to choose the first vertex, the *source* **S**, and the last vertex, the *destination* **D**, to be traversed. Three choices are especially useful (see Fig. 3), these three types of blocks will be served as tiling blocks:

Type I: **S** is the bottom-right corner string, **D** is the upper-right corner string.

Type II: **S** is the upper-left corner string, **D** is the upper-right corner string.

Type III: **S** is the bottom-right corner string, **D** is the bottom-left corner string.

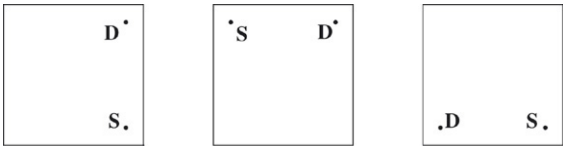

Fig. 3. Type I, II, III choices for the source and the destination for a block.

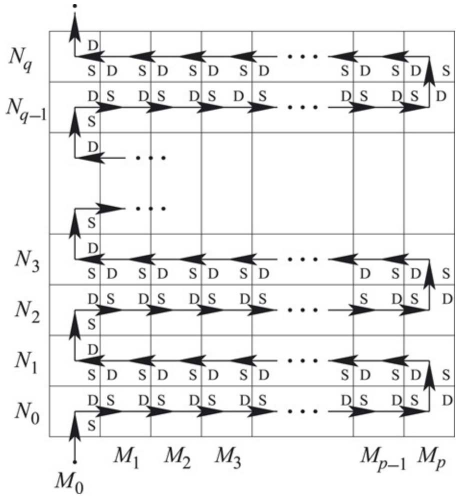

Fig. 4. Generate an $(m+n, 2t)$ code from an (m, t) code and an (n, t) code.

Now consider the Cartesian product $(M_0, M_1, \ldots, M_p) \times (N_0, N_1, \ldots, N_q)$, consisting of pq blocks $M_i N_j$, $0 \le i \le p, 0 \le j \le q$. We traverse all blocks by starting from the block $M_0 \times N_0$, and zigzaging in the order

$$\begin{array}{lllll}
M_0 \times N_0, & M_1 \times N_0, & M_2 \times N_0, & \ldots, & M_p \times N_0, \\
M_p \times N_1, & M_{p-1} \times N_1, & M_{p-2} \times N_1, & \ldots, & M_0 \times N_1, \\
M_0 \times N_2, & M_1 \times N_2, & M_2 \times N_2, & \ldots, & M_p \times N_2, \\
 & & \vdots & & \\
M_0 \times N_{q-1}, & M_1 \times N_{q-1}, & M_2 \times N_{q-1}, & \ldots, & M_p \times N_{q-1}, \\
M_p \times N_q, & M_{p-1} \times N_q, & M_{p-2} \times N_q, & \ldots, & M_p \times N_q.
\end{array}$$

We do not proceed to next block until all strings in the same block are traversed. This is guaranteed as above and by suitably arranging Type I, II, III blocks as tiling blocks, making the destination **D** of the current block and the source **S** of the next block differ at only one bit. An explicit construction assigns Type I to blocks $M_0 \times N_j$ for every $0 \le j \le q$, Type II to blocks with $j \equiv 0 \bmod 2$ and $1 \le i \le p$, and Type III to blocks with $j \equiv 1 \bmod 2$ and $1 \le i \le p$, see Fig. 4. Since q is odd, the destination **D** of the block $M_0 \times N_q$ and the source **S** of the block $M_0 \times N_0$ also differ only at one bit. In this way we can traverse all 2^{m+n} strings, with each string $2t$ steps away from its complement. That is, we have constructed an $(m+n, 2t)$-antipodal Gray code. □

Consequently, we have an alternative proof for Killian and Savage's result [11].

Theorem 18 (*Killian and Savage [11]*). *For any positive integer $k \ge 1$, there is a simple $(2^k, 2^k)$-antipodal Gray code.*

Proof. The theorem follows from Theorem 17 and the fact that $(00, 01, 11, 10)$ is a simple $(2, 2)$-antipodal Gray code. □

We in fact can get more simple (n, t)-antipodal Gray codes by similar arguments. For instance, from a simple $(4, 8)$-antipodal Gray code, we can construct simple $(2^k, 2^{k+1})$-antipodal Gray codes for $k \ge 2$; from a simple $(6, 8)$-antipodal Gray code, we can construct simple $(3 \cdot 2^k, 2^{k+2})$-antipodal Gray codes for $k \ge 1$. More can be constructed if Theorem 13 is used.

5. Conclusion remarks

Even taking all results in the previous section into consideration, for even n we still know very little about the existence of (n, t)-antipodal Gray codes. Since (2, 2)-, (4, 4)-, (4, 6)-, and (4, 8)-antipodal Gray codes do exist, the cases $n = 2, 4$ are completely solved. However, even for $n = 6$, we have only partial results, namely the existence of (6, 8)-, (6, 16)-, (6, 24)-, (6, 30)-, and (6, 32)-antipodal Gray codes, let alone larger n. The non-existence of the (6, 6) code is done in [11], hence it might not be anticipated that the codes exist for all other pairs of (n, t). Three next unknown cases are (6, 10), (8, 10) and (10, 10). A possible direction is to design new methods for constructing longer codes from known ones to fill in the gap as many as possible.

Acknowledgements

The authors thank the referee for many constructive suggestions.

First author was partially supported by the National Science Council under grant NSC 95-2221-E-002-125-MY3. Second author was partially supported by the National Science Council under grant NSC 95-2115-M-390-006-MY3 and TJ&MY Foundation.

References

[1] D.J. Amalraj, N. Sundararajan, G. Dhar, A data structure based on gray code encoding for graphics and image processing, in: Applications of Digital Image Processing XIII, in: SPIE, vol. 1349, SPIE, Bellingham, MA, 1990.
[2] B. Bultena, F. Ruskey, Transition restricted Gray codes, Electron. J. Combin. 3 (1) (1996). Research Paper 11, 11pp. (electronic).
[3] C.-C. Chang, H.-Y. Chen, C.-Y. Chen, Symbolic gray code as a data allocation scheme for two-disc systems, Comput. J. 35 (3) (1992) 299–305.
[4] M.-S. Chen, K.-G. Shin, Subcube allocation and task migration in hypercube multiprocessors, IEEE Trans. Comput. 39 (9) (1990) 1146–1155.
[5] P. Diaconis, S. Holmes, Gray codes for randomization procedures, Stat. Comput. (4) (1994) 207–302.
[6] C. Faloutsos, Gray codes for partial match and range queries, IEEE Trans. Softw. Eng. 14 (10) (1988) 1381–1393.
[7] M. Gardner, Mathematical games: The curious properties of the Gray code and how it can be used to solve puzzles, Scientific Amer. 227 (1972) 106–109.
[8] E.N. Gilbert, Gray codes and paths on the n-cube, Bell System Tech. J. 37 (1958) 815–826.
[9] L. Goddyn, G.M. Lawrence, E. Nemeth, Gray codes with optimized run lengths, Util. Math. 34 (1988) 179–192.
[10] F. Gray, Pulse code communication, US Pateht 2632058, March 1958.
[11] C.E. Killian, C.D. Savage, Antipodal Gray codes, Discrete Math. 281 (2004) 221–236.
[12] D.E. Knuth, Pre-fascicle 2a: Generating all n-tuples, a preview of section 7.2.1.1, in: The Art of Computer Programming, vol. 4, http://www-cs-faculty.standford.edu/˜knuth/news.html.
[13] R.M. Losee, A gray code based ordering for documents on shelves: Classification for browsing and retrieval, J. Amer. Soc. Inform. Sci. 43 (4) (1992) 312–322.
[14] J.E. Ludman, Gray code generation for MPSK signals, IEEE Trans. Commun. COM-29 (1981) 1519–1522.
[15] A. Nijenhuis, H.S. Wilf, Combinatorial Algorithms, second ed., Academic Press, 1978.
[16] M. Ramras, A new method of generating Hamiltonian cycles on n-cube, Discrete Math. 85 (3) (1990) 329–331.
[17] D.S. Richards, Data compression and Gray-code sorting, Inform. Process. Lett. 22 (4) (1986) 201–205.
[18] J.P. Robinson, M. Cohn, Counting sequences, IEEE Trans. Comput. 30 (1) (1981) 17–23.
[19] C. Savage, A survey of combinatorial Gray codes, SIAM Rev. 39 (4) (1997) 605–629.
[20] C.D. Savage, P. Winkler, Monotone Gray codes and the middle levels problem, J. Combin. Theory, Ser. A 70 (2) (1995) 230–248.
[21] V.E. Vickers, J. Silverman, A technique for generating specialized Gray code, IEEE Trans. Comput. 29 (4) (1980) 329–331.